

Administration unter Ubuntu

Wechsel zu Ubuntu 18.04 - Eine Einrichtung für Free Pascal Entwickler

Von Anfang an stand bei mir fest das ich umsteigen wollte, alleine schon wegen der Sytemupdates unter Windows 10 die mir immer wieder Probleme bereiteten. Immer wieder hatte ich einen Umstieg auf Linux angegangen und immer wieder bin ich in letzter Konsequenz daran gescheitert für alle meinerseits benötigten Einsatzgebiete passende alternativen zu finden, doch diese Mal kam es anders.

Client Installation

Zuerst hatte ich mir eine Liste zusammen gestellt was meine Mindestanforderungen an meinen Entwicklungsrechner mit seinem System und der Entwicklungsumgebung sind.

- ein System das ein besseres Rechtesystem hat, aber nicht für den Nutzer zu kompliziert wird
- Firefox als Browser, unter Ubuntu Standard, unter Windows immer noch meine erste Wahl als Entwickler
- Thunderbird als E-Mail-Client, unter Ubuntu Standard, unter Windows mein bisheriger E-Mail-Client
- Lazarus / Free Pascal (FPCUPdeluxe) läuft unter Ubuntu wie auch unter Windows
- Remote Desktop Betrieb zum Server (nicht nur Konsole) unter Ubuntu über Remmina
- Dateifreigabe im Netzwerk, unter Ubuntu in Nautilus über Samba
- SVN und Git, unter Ubuntu Git Cola/DAG (in Nautilus mit RabbitVCS), als Alternative zu Windows im Explorer mit TortoiseSVN und TortoiseGit
- ein Ersatz für Notepad++, unter Ubuntu kein equivalentes Gegenstück aber als teilweiser Ersatz SciTE, GEdit, GHex, Meld, ...
- ein Ersatz für Paint.net, unter Ubuntu Pinta
- Inkscape läuft unter Ubuntu wie auch unter Windows

Sicherlich gibt es einige Wege um ein Entwicklungsrechner aufzusetzen, aber gerade Umsteiger von Windows haben es am Anfang nicht gerade leicht mit Ubuntu.

Gestartet bin ich mit einem USB-Stick mit Ubuntu 18.04 Bionic Beaver und der Standard Installation, dieses Vorgehen möchte ich nicht weiter beschreiben denn hiervon gibt es zahlreiche Beschreibungen im Netz.

Starten möchte ich den anschließenden Schritten, der Installation und Konfiguration der Komponenten.

Tritt in den folgenden Passagen [user] im Text oder Code auf ist an dieser Stelle der eigene Benutzer gemeint.

Vorbereitungen

Nautilus um die Funktion "Neue Dokument" anlegen erweitern, im Verzeichnis "/home/[user]/Vorlagen" eine Datei leer.txt anlegen

```
cd Vorlagen  
touch leer.txt
```

Unter Ubuntu werden viele Vorgänge im Terminal-Fenster durchgeführt und oft sind es Befehle für die man Administrationsrechte benötigt also "SUDO", was ist da naheliegender als sich ein Terminal-Fenster mit

sudo-Rechten zu öffnen damit man nicht vor jeden Befehl sudo tippen muss.

Terminal Fenster mit root Rechten (sudo)

```
sudo -i  
[sudo] Passwort for root:
```

Einmal das Passwort eingeben für alle folgenden Befehle.

Ein Update der Paketliste durchführen

```
apt-get update  
apt-get upgrade
```

Nautilus um die Dateibehandlung "Als Administrator bearbeiten" erweitern, damit das Bearbeiten von geschützten Dateien vereinfacht wird

```
apt install nautilus-admin  
nautilus -q
```

Entwicklungsumgebung installieren

Die fehlenden Pakete für Free Pascal und Lazarus nachinstallieren

```
apt-get install libx11-dev  
apt-get install libgdk-pixbuf2.0-dev  
apt-get install libpango1.0-dev  
apt-get install libgtk2.0-dev
```

Möchte man mit OpenGL (TOpenGLControl) entwickeln ist auch noch das folgende Paket erforderlich

```
apt-get install freeglut3-dev
```

SVN / Git installieren

```
apt-get install subversion  
apt install git
```

SVN / Git GUI installieren

```
apt-get install rabbitvcs-cli rabbitvcs-core rabbitvcs-gedit rabbitvcs-  
nautilus  
apt install git-cola
```

Free Pascal und Lazarus installieren

fpcupdeluxe-x86_64-linux auf <https://github.com/LongDirtyAnimAlf/fpcupdeluxe/releases>
(<https://github.com/LongDirtyAnimAlf/fpcupdeluxe/releases>) herunterladen und im persönlichen Ordner (/home/[user]/) unter fpcupdeluxe ablegen und in Nautilus mit Rechts-Klick auf die Datei, die Eigenschaften bearbeiten, im Reiter Zugriffsrechte Ausführen Haken setzen. (Properties-->Permissions-->Allow executing file as program). Nun lässt sich das Programm einfach starten, im folgenden habe ich einfach das Installations-Verzeichnis im Benutzerverzeichnis /home/[user]/fpcupdeluxe angegeben und sowohl Free Pascal als auch Lazarus in der Trunk Version installiert (dieses lief unter Ubuntu wesentlich schneller als unter Windows).

Damit man in Zukunft dieses Programm zum Updaten von Free Pascal als auch Lazarus benutzen kann empfiehlt es sich eine Verknüpfungsdatei anzulegen (desktop shortcut launcher)

Eine Datei FPCUPdeluxe.desktop im eigenen persönlichen Ortner (/home/[user]/) unter .local/share/applications anlegen (das Icon habe ich mir aus dem fpcupdelux Projekt besorgt):

```
[Desktop Entry]
Version=1.0
Encoding=UTF-8
Type=Application
Icon=/home/[user]/fpcupdeluxe/fpcupdeluxe.ico
Exec=/home/[user]/fpcupdeluxe/fpcupdeluxe-x86_64-linux
Name=FPCUPdeluxe
GenericName=GUI to install an update Lazarus IDE and Free Pascal Compiler
Categories=Application;Development;GUIInstaller;
Keywords=editor;Pascal;IDE;FreePascal;fpc;Design;Designer;
Name[de_DE]=FPCUPdeluxe
```

Um diese Datei startbar zu machen auch hier in Nautilus mit Rechts-Klick auf die Datei, die Eigenschaften bearbeiten, im Reiter Zugriffsrechte Ausführen Haken setzen.

Falls fpcdeluxe es immer noch nicht schafft eine funktionierende Verknüfungsdatei anzulegen folgt hier nun noch die passende Datei für Lazarus

Eine Datei Lazarus.desktop im eigenen persönlichen Ortner (/home/[user]/) unter .local/share/applications anlegen:

```
[Desktop Entry]
Version=1.0
Encoding=UTF-8
Type=Application
Icon=/home/[user]/fpcupdeluxe/lazarus/images/icons/lazarus.ico
Exec=/home/[user]/fpcupdeluxe/lazarus/lazarus --
pcp="/home/[user]/fpcupdeluxe/config_lazarus" %f
Name=Lazarus_fpcupdeluxe
GenericName=Lazarus IDE with Free Pascal Compiler
Categories=Application;IDE;Development;GUIDesigner;
Keywords=editor;Pascal;IDE;FreePascal;fpc;Design;Designer;
Name[de_DE]=Lazarus
```

Nun sollten sowohl FPCUPdeluxe als auch Lazarus in der Anwendungsliste (Anwendungen anzeigen) angezeigt werden, dort sind sie auch aufrufbar, ich habe sie mir direkt zu den Favoriten hinzugefügt.

Tools zur Bildbearbeitung/Optimierung installieren

libpodofo-utils um uncompress von pdf zu ermöglichen

```
apt install libpodofo-utils
```

jpegoptim und jpegtran zur jpeg Optimierung

```
apt install jpegoptim
apt install libjpeg-progs
apt install libjpeg-turbo-progs
```

optipng zur png Optimierung

```
apt install optipng
```

gifsicle zur gif Optimierung

```
apt install gifsicle
```

Erweiterung für Nautilus zum anzeigen von Video-Thumbnails

```
apt install ffmpegthumbnailer
```

Allgemeiner mp4 support

```
apt install libdvdnav4 libdvdread4 gstreamer1.0-plugins-bad gstreamer1.0-  
plugins-ugly libdvd-pkg  
apt install ubuntu-restricted-extras
```

WebP support

```
apt install webp
```

pdftohtml - poppler 0.62.0 braucht nicht installiert werden

Installieren der paint.net alternative pinta und inkscape als Bild/Vektor-Bearbeitungsprogramme

```
apt install pinta  
apt install inkscape
```

Erweiterung für Nautilus zum Reduzieren der Größe von JPG/PNG Dateien

```
add-apt-repository ppa:atareao/nautilus-extensions  
apt update  
apt install nautilus-reduceimages  
nautilus -q
```

Komprimierungs-Tools

zlib

```
apt install build-essential checkinstall zlib1g-dev -y
```

7zip

```
apt install p7zip-full
```

brothli

```
apt install brotli
```

OpenSSL aus den Quellen erzeugen

OpenSSL download der Sourcen

```
cd /usr/local/src/  
wget https://www.openssl.org/source/openssl-1.1.1.tar.gz  
tar -xf openssl-1.1.1.tar.gz
```

OpenSSL Installation

```
cd openssl-1.1.1  
./config --prefix=/usr/local/ssl --openssldir=/usr/local/ssl shared zlib  
make  
make install
```

link libraries konfigurieren

Datei anlegen /etc/ld.so.conf.d/openssl-1.1.1.conf (in Nautilus "Als Administrator öffnen", Neues Dokument leer, "Als Administrator bearbeiten", im KEdit speichern unter)

```
cd /etc/ld.so.conf.d
touch openssl-1.1.1.conf

/usr/local/ssl/lib
```

Datei /etc/manpath.config bearbeiten (in Nautilus "Als Administrator bearbeiten") und die folgende Zeile einfügen

```
MANPATH_MAP /usr/local/ssl/bin /usr/local/ssl/man
```

manual database Update

```
mandb
```

Datei /etc/environment (in Nautilus "Als Administrator bearbeiten") /usr/local/ssl/bin hinzufügen

```
PATH="/usr/local/sbin:/usr/local/bin:/usr/local/ssl/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games"
```

Erfolgreiche Änderung überprüfen

```
ldconfig -v
which openssl
openssl version -a
```

Um etwas mehr an den Desktop Einstellungen verändern zu können oder sogar ein Windows ähnlicheres Aussehen zu erreichen kann man ein paar nützliche Tools installieren

dconf installieren

```
apt install dconf-tools
```

Windows look

```
apt install gnome-shell-extensions gnome-shell-extension-dash-to-panel gnome-tweaks
```

restart

dconf -> Tweaks: Optimierungen -> Erweiterungen -> Dash to Panel

Firefox und Thunderbird

Das Firefox Profile welches ich unter Windows gesichert habe wie auch das Thunderbird Profile in den Ordner (/home/[user]/) unter .mozilla/firefox/[profilename] bzw. .thunderbird/[profilename] ablegen, falls die Ordner noch nicht existieren einfach die Anwendung einmal starten.

Im Termninal lassen sich die beiden Anwendungen mit Übergabeparameter -p aufrufen danach erscheint ein Dialog zur Profile-Auswahl und zur Einstellung des Standard-Profiles.

```
firefox -p
thunderbird -p
```

Dateifreigabe über Samba

```
apt install system-config-samba
```

Datei /etc/libuser.conf anlegen (in Nautilus "Als Administrator öffnen", Neues Dokument leer, "Als Administrator bearbeiten", im KEdit speichern unter)

```
cd /etc  
touch libuser.conf
```

Grafisches Benutzerinterface für Samba

```
system-config-samba
```

Samba Neustart

```
systemctl restart smbd.service
```

Tools

SciTE

```
apt-get install scite
```

GHex Editor

```
apt-get install ghex
```

Bei Einrichtung eines Dual-Boot Systems ist eine ntfs Partition zum Datenaustausch zwischen den Systemen sinnvoll (ntfs data partition mount)

Datei /etc/fstab (in Nautilus "Als Administrator bearbeiten" im KEdit nach der ntfs Partition suchen und die Zeile bearbeiten)

```
UUID=342708ED2C7A02F2 /home/udo/data ntfs  
defaults,nls=utf8,umask=000,dmask=027,fmask=137,uid=1000,gid=1000,windows_names  
0 0
```

Fingerprint Scanner unterstützung einrichten

```
apt install -y fprintd libpam-fprintd
```

Datei /etc/pam.d/common-auth (in Nautilus "Als Administrator bearbeiten" im KEdit folgende Zeile einfügen)

```
auth [success=2 default=ignore] pam_fprintd.so max_tries=1 timeout=10
```

System aufräumen

alte Kernel entfernen

```
apt autoremove
```

überflüssige Abhängigkeiten entfernen

```
apt-get autoremove
```

Paket-Cache leeren

```
apt-get clean
```

Server Installation

Für den Server kann man natürlich ein spezielles Ubuntu Server Image herunterladen, da ich aber eine grafische Benutzeroberfläche bevorzuge auf die ich Remote zugreifen kann habe ich das Selbe Image wie zur Client Installation genommen, allerdings nur die minimale Installation durchgeführt.

Nach der Installation habe ich die selben Vorbereitungen wie bei der Client Installation durchgeführt

Installation des Remote Desktop Freigabe

```
apt update && sudo apt install -y vino
apt -y install xrdp
systemctl enable xrdp
sudo ufw allow 3389/tcp
sudo /etc/init.d/xrdp restart
```

Unter Einstellungen -> Freigabe sowohl in der Kopfleiste Einschalten als auch die Bildschirmfreigabe aktivieren, hier das passende Netzwerk auswählen, Verbindungen zur Steuerung des Bildschirms erlauben und ein Passwort vergeben.

Mit Remmina auf die Remote Desktop Freigabe zugreifen

In Remmina über den Plusbutton ein neues Verbindungsprofil erstellen, Protokoll VNC auswählen, Bezeichnung vergeben, bei Server den Rechnernamen und bei Benutzerpasswort das Passwort der Freigabe eingeben, anschließend speichern.

Je nach gewünschter Qualität, Einschränkungen und bezüglich eines SSH-Tunnel können auch noch Einstellungen vorgenommen werden, die sind aber nicht unbedingt erforderlich.

Dateifreigabe über Samba

Die Dateifreigabe wie bei der Client Installation durchgeführt.

Installation des eigenen Servers oder LAMP (Apache, PHP 7.2 und MySQL) Installation

```
apt-get install apache2 libapache2-mod-php7.2 php7.2 php7.2-mysql mysql-server
```

Installation nützlicher Netzwerktools

```
apt install net-tools
```

Hosts Entry des Services in /etc/hosts eintragen

```
127.0.0.1 www.gocher.me
```

dhcp freigabe ipv6

```
apt install dhcpcd5
```

Datei /etc/dhcpd.conf bearbeiten (in Nautilus "Als Administrator bearbeiten") und folgende Zeile hinzufügen

```
slaac hwaddr
```

Einstellungen überprüfen

```
ifconfig
```

Eine Service für den eigens geschrieben daemon anlegen, hier am Beispiel meines Webservers (httpServer.service)

Das Verzeichnis /etc/systemd/system (in Nautilus "Als Administrator bearbeiten")
/etc/systemd/system/httpServer.service hinzufügen

```
[Unit]
Description=Udos Webserver
After=network-online.target
[Install]
[Service]
#User=udo
# The configuration file application.properties should be here:
Type=simple
#path of workspace
WorkingDirectory=/home/[user]/webserver
#path to executable.
ExecStart=/home/[user]/webserver/httpServer -r
#Restart=on-failure
[Install]
WantedBy=multi-user.target
```

Zum Ausführen hier noch die passenden Komandos

```
systemctl daemon-reload
systemctl enable httpServer.service
systemctl start httpServer.service
systemctl status httpServer.service
systemctl reload-or-restart httpServer.service
systemctl stop httpServer.service
systemctl list-units
```

Let's encrypt certbot installieren und einrichten

```
add-apt-repository ppa:certbot/certbot
apt update
dpkg --configure -a
apt install certbot
```

Certificate erzeugen

Hier wieder am Beispiel von meinem Webauftritt.

```
certbot certonly --agree-tos --email udo.schmal@t-online.de --webroot -w  
/home/udo/webserver/__ssl/ -d gocher.me -d www.gocher.me -d h2.gocher.me -d  
maps.gocher.me -d test.gocher.me -d app.gocher.me
```

result is in /etc/letsencrypt

Autor: [Udo Schmal](#), veröffentlicht: 10.03.2018, letzte Änderung: 29.08.2024

[© Copyright 2024 Udo Schmal](#)